Práctica: Cifrado simétrico de un fichero

Objetivo: Realizar el cifrado de un fichero con un algoritmo simétrico, derivando la clave de cifrado a partir de una password

Desarrollo:

1. Crear un clase para llevar a cabo el cifrado y descifrado
   1.1. Método de cifrado

```
public static void cifrarDES(String fileIn, String fileOut, String clave)   throws Exception {
        // Semilla a partir del generador de numeros aleatorios
        byte[] salt = new byte[8];
        SecureRandom random = new SecureRandom();
        random.nextBytes(salt);

        // Derivar la clave de cifrado a partir de la semilla y password
        SecretKeyFactory factory = SecretKeyFactory.getInstance("PBEWithMD5AndDES");
        PBEKeySpec spec = new PBEKeySpec(clave.toCharArray(), salt, 1);
        SecretKey sKey = factory.generateSecret(spec);

        // Generar el cifrador
        Cipher cipher = Cipher.getInstance(sKey.getAlgorithm());
        cipher.init(Cipher.ENCRYPT_MODE, sKey, new PBEParameterSpec(salt, 1));

        // Leer el fichero
        byte[] fileContent = getBytesFromFile(fileIn);

        // EscribirFichero
        OutputStream writer = new FileOutputStream(fileOut);
        writer.write(salt);
        writer.write(cipher.doFinal(fileContent));
        writer.close();
}
```

   1.2. Método de descifrado

```
public static void descifrarDES(String fileIn, String fileOut, String clave) throws Exception {
        // Leer fichero y extraer la información
        byte[] fileContent = getBytesFromFile(fileIn);
        byte[] salt = new byte[8];
        byte[] content = new byte[fileContent.length - 8];

        for (int i = 0; i < 8; i++) {
                salt[i] = fileContent[i];
        }

        for (int i = 8; i < fileContent.length; i++) {
                content[i - 8] = fileContent[i];
        }

        // Derivar la clave de cifrado a partir de la semilla y password
        SecretKeyFactory factory = SecretKeyFactory.getInstance("PBEWithMD5AndDES ");
        PBEKeySpec spec = new PBEKeySpec(clave.toCharArray(), salt, 1);
        SecretKey sKey = factory.generateSecret(spec);

        // Generar el cifrador
        Cipher cipher = Cipher.getInstance(sKey.getAlgorithm());
        cipher.init(Cipher.DECRYPT_MODE, sKey, new PBEParameterSpec(salt, 1));

        // EscribirFichero
        OutputStream writer = new FileOutputStream(fileOut);
        writer.write(cipher.doFinal(content));
        writer.close();
}
```

Web: www.sidertia.com
Email: info@sidertia.com
Tel: 646.366.781 | 646.116.305
Centro de Empresas Carabanchel
C/ Cidro, 3 - Oficina 207 28044 MADRID

1.3. Método auxiliar para cargar los datos de un fichero

```
private byte[] getBytesFromFile(String fileIn) throws IOException {
        File file = new File(fileIn);

        InputStream is = new FileInputStream(fileIn);
        long length = file.length();

        byte[] bytes = new byte[(int) length];

        int offset = 0;
        int numRead = 0;
        while (offset < bytes.length && (numRead = is.read(bytes, offset, bytes.length - offset)) >= 0) {
                offset += numRead;
        }

        is.close();
        return bytes;
}
```

2. Cifrar y descifrar los ficheros PDFs y comprobar que todo funciona correctamente
3. Intentar descifrar los ficheros con CrypTool

Web: www.sidertia.com
Email: info@sidertia.com
Tel: 646.366.781 | 646.116.305
Centro de Empresas Carabanchel
C/ Cidro, 3 - Oficina 207 28044 MADRID